(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 March 2001 (29.03.2001)

PCT

(10) International Publication Number
WO 01/22373 A1

(51) International Patent Classification[7]: G07F 7/10, 19/00, H04L 9/32

(21) International Application Number: PCT/NL00/00217

(22) International Filing Date: 3 April 2000 (03.04.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/NL99/00590
22 September 1999 (22.09.1999) NL

(71) Applicant *(for all designated States except US)*: **BA CARDS AND SECURITY B.V. (BACS)** [NL/NL]; P.O. Box 28024, NL-3828 ZG Hoogland (NL).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **MOSTERD, Michiel** [NL/NL]; Wilhelminastraat 34/3, NL-1054 WJ Amsterdam (NL). **KROP, Hildebrand, Ruben** [NL/NL]; Nova Zemblastraat 451, NL-1013 RJ Amsterdam (NL). **HEIDEN, Paul, Richard** [NL/NL]; Maasbandijk 15, NL-5332 KB Kerkdriel (NL). **VAN WACHEM, Paul, Anthony** [NL/NL]; Julianalaan 185, NL-3722 GK Bilthoven (NL).

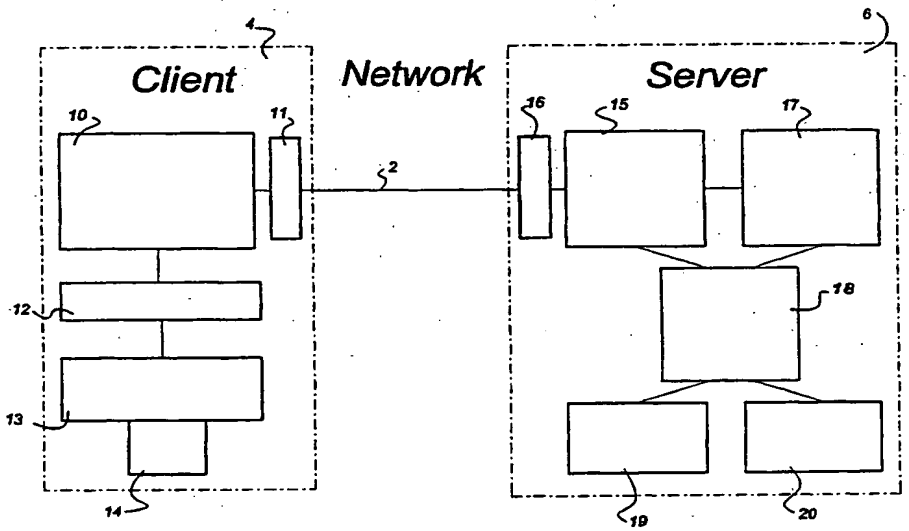
(74) Agent: **JORRITSMA, Ruurd**; Nederlandsch Octrooibureau, Scheveningseweg 82, P.O. Box 29720, NL-2502 LS The Hague (NL).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR PERFORMING A TRANSACTION BETWEEN A CLIENT AND A SERVER OVER A NETWORK



(57) Abstract: Method and system for performing a transaction between a first client (4) and a server (6) over a network (2), including the steps of providing a first signed record by digitally signing a record by the first client (4), the record comprising information to be accepted by the first client (4). Furthermore, a second signed record is provided by digitally signing the record by the server (6), which is sent to the first client (4). The information is stored both by the server (6) and the first client (4), thereby providing proof of the transaction. Furthermore, the server (6) may sign for one or more parties (4, 26) by proxy, based on a certificate. Also, the method and system may be used to provide fair exchange of signatures of the parties (4, 26) involved, by using escrows of the signatures required.

WO 01/22373 A1

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Method and system for performing a transaction between a client and a server over a network

The present invention relates to a method for performing a transaction between a first client and a server over a network, including the steps of providing a server authentication to the first client, providing a first client authentication to the server, sending a record from the server to the first client via the network, the record comprising information to be accepted by the first client, providing a first signed record by digitally signing the record by the first client, sending a first message from the first client to the server via the network, the first message at least comprising the first signed record, and storing the record, first message and the first client authentication by the server upon positive verification of the first signed record with the first client authentication. The network may be an untrusted network.

Such a method is known from US-A-5,850,442, which describes a method for performing secure electronic transactions between two parties, e.g. clients and servers, via a network. The electronic transactions can be securely performed using a public key infrastructure and smart token technology, such as smart cards.

Information originating from one of the two parties is authenticated as to origin, using encryption keys from a smart token. One of the applications disclosed is the placing of an order. A client connects to a server via the network and obtains an order form for placing an order with the server. The client digitally signs the order and sends it to the server, which may validate the origin of the digitally signed order by checking the signature by means of the encryption keys from the smart token. Another application described is a method for performing an electronic money withdrawal. An electronic withdrawal slip is electronically signed by a customer and transferred to a bank, which stores the signed slip. In return, the bank prepares an electronic cash certificate, digitally signs it and sends it to the customer, which may store the cash certificate. This method is in fact a series combination of two subsequent methods as described above and involves two different groups of information.

A disadvantage in this and other present day methods and systems for performing secure electronic transactions via a network is that no proof of the transaction can be made by at least one of the parties involved. Usually only one party keeps a verifiable

record of the transaction. Furthermore, the records stored by one of the parties may be tampered with.

The object of the present invention is to provide a method for performing secure electronic transactions via a network, in which both parties obtain solid proof of the transaction.

This object is achieved by a method as defined in the preamble, in which the method further comprises the steps of providing a second signed record by digitally signing the record by the server, sending a second message to the first client, the second message at least comprising the second signed record, and storing the record, the second message and the server authentication by the first client upon verification of the second signed record with the server authentication.

The method according to the present invention has the advantage that both the first client and the server have stored the record and the respective signed records and authentication. At all times after the transaction, both the server and the first client are able to verify and provide proof of the transaction made. Tampering with information in the record describing the transaction can be detected by both the first client and the server by checking the digitally signed records.

In a further embodiment, the first client authentication further comprises an application serial number, and the method further comprises the steps of computing a unique application key from the application serial number and a master key by the server, using the unique application key to encrypt the second message by the server before sending the second message to the first client, and decrypting the second message by the first client. The application serial number may, e.g., be a smart card serial number. This enables the storing of the record and other data by the first client to be write-protected. Only a server that has a correct master key and the proper application serial number, as received from the first client in the first message, is able to generate the appropriate unique application key. The first client also has the unique application key, and is able to decrypt the second message properly and store the decrypted data.

Preferably, in a further embodiment, the first message further comprises a random number, and the method further comprises the step of performing an exclusive-or-operation by the server on the second message using the random number before sending the second message to the first client. In this case, an exclusive-or-operation

with the same random number resulting in an unscrambled second message can only be performed by the first client who sent that second message and knows the random number. This embodiment ascertains that a certain second message can only be stored once by the first client.

In a still further embodiment, the step of providing a second signed record comprises the steps of computing a hash of the record and digitally signing the hash of the record by the server. This embodiment enables to send a second message with only a limited size, rendering advantages with respect to transfer times and required memory size.

The step of digitally signing the record by the server may be performed using an additional secret server key, the server authentication further comprising an additional public server key.

In a further embodiment, the first and second message further comprises a file address for storing the second message by the first client. Storage by the first client may be done on, e.g. a smart card with only a limited memory size. This embodiment enables to attach a file address on the smart card to a certain transaction. When the memory of the smart card is full, a previously stored record may be replaced by the new record, replacing e.g. the oldest record or the record with the lowest priority.

In a further embodiment, the second message further comprises a unique record number, and the method further comprises the step of sending a third message to the server by the first client upon positively verification of the content of the second message, the third message comprising the unique record number.

This enables the server to verify that the second message was received correctly by the first client and that the first client has obtained proof of the transaction. When no third message is received, the server can try to contact the first client again at a later time.

An embodiment of the method uses a record, in which the record comprises unformatted text, such as a plain text document, or a plain text description of the document format. Graphical user interfaces may show information in a manner invisible or unclear to one of the parties (client or server), enabling the possibility that one of the parties digitally signs an order comprising information the party concerned does not want to sign when presented to one of the parties for visual verification before digitally signing the record. This embodiment greatly reduces the chance of tampering

with the information in the record when using graphical user interfaces to show the record to the first client. Preferably, the textual content of the record comprises, next to possible video and/or audio fragments, unformatted text only.

Preferably, the step of providing server authentication and first client authentication is obtained by means of RSA-keys and X.509 certificates. Methods using RSA-keys and X.509 certificates have proven to be safe in use and are wide spread. Another possible method is the DSS algorithm. Preferably, these are based on large prime numbers.

In a further embodiment of the present invention the second signed record is provided by the server by digitally signing the record using a signing key from a second client, the signing key allowing transactions between the second client and at least the first client. Preferably, the signing key is formed by a certificate signed by the second client, the certificate being received from the second client and stored by the server.

In this embodiment, it is possible that the server signs by proxy for one or both parties to the transaction. A sales manager may, e.g., authorise the company server to sign purchase orders from one or more customers. This is preferably done by means of a certificate, allowing deals to be made automatically between the company server and one or more customers, while the manager remains responsible for these deals, and not the IT department.

A further embodiment of the method according to present invention comprises the further steps of sending a non-verifiable escrow of the second signature by the server to the first client before sending the second message, and sending a verifiable escrow of the first signature by the first client to the server before sending the first message.

This makes it possible that either both parties end up with each other's signatures, or that no party to the transaction has the opposite party's signature. This can be especially important in contract negotiations where both parties want the other party to sign first. This embodiment does however require more data traffic and data processing.

The first client can simply abort the transaction in a first instance, by not reacting further when the record, comprising information to be accepted by the first client is received. When the first client has sent the verifiable escrow, the transaction can be aborted by the server when the verifiable escrow of the first signature is incorrect.

When further on in the transaction process, a signature turns out to be incorrect, both parties can obtain the opposite party's signature by means of resolving the transaction, e.g. by means of a trusted third party which provides the second signature associated with the non-verifiable escrow to the first client or the first signature associated with the verifiable escrow to the server.

In a second aspect, the present invention relates to a system for performing a secure transaction over a network, comprising at least one server computer, at least one client computer and a network, the network providing a connection between the at least one server computer and the at least one client computer, the server computer and client computer being arranged to execute the respective server and first client steps of the method according to one of the method claims of the present invention. Preferably, the client computer further comprises memory interface means for communicating with smart memory means, the smart memory means being arranged to execute at least one of the steps of the method according to one of the method claims according to the present invention which are performed by the first client.

Furthermore, the present invention relates to a computer program comprising computer program code, which in operation provides a computer system with the functionality of the method according to the present invention. The present invention also relates to a computer program product comprising such a computer program.

The present invention will now be described in more detail with reference to the accompanying drawings, in which

Fig. 1 shows a schematic view of a system for performing a secure transaction between a server computer and a client computer over a network;

Fig. 2 shows a schematic view of an alternative system according to the present invention; and

Fig. 3 shows a flow diagram of an embodiment of the method according to the present invention.

The present invention provides a method for conducting transactions over a network, more specifically over the Internet and a system for performing the present method.

Fig. 1 shows a schematic view of a network 2, such as the Internet, in which two parties are shown being connected to each other via the network 2. A first party offers a product or service via the network 2, the first party being connected to the network 2 by

a server computer 6. The other party may be willing to buy a product or service from the first party and is connected to the network 2 by a client computer 4.

The network 2 may be any open network, such as the Internet, which provides an unsecured connection for exchange of data or information between the client 4 and the server 6.

The client computer 4 is connected to the network 2 by a client network interface 11, which may be arranged to establish a Secure Socket Layer connection with the network 2. Furthermore, the client computer 4 is equipped with a web browser application 10 to retrieve information from the network 2 and a transaction client service, further called Q™ client service 12 implementing the client part of the method according to the present invention. Furthermore, the client computer 4 comprises a smart card reader 13 to exchange data with a smart card 14.

The server computer 6 is equipped with a web server 15, providing information about the product and services for sale onto the network 2. The web server 15 is connected to the network 2 via a server network connection 16, which may also be equipped to provide a Secure Socket layer connection. Furthermore, the server computer 6 may be arranged to host further services communicating with the network 2 via the web server 15, including an order application 17 and a transaction service, further called Q™ service 18. The order application 17 processes orders received from the client computer 4 via the network 2. The Q™ service 18 is arranged to provide mutual proof of a transaction via the network 2. The Q™ service 18 may be exchanging information with a Q™ order database 19 and with a Q™ Security Access Module (SAM) 20. It will be clear that the Q™ service 18, Q™ order database 19 and Q™ SAM 20 may be located on a separate Q™ server computer 25 (as e.g. shown in Fig. 2) which is connected to the server computer 6, which in that case hosts the web server 15 and order application 17.

The method according to the present invention assures that, after the transaction is confirmed, both the server and the client have evidence of this event. The client has a receipt stored on the smart card 14 and the server stores the proof of transaction in his database 19. Neither the server nor the client will be able to modify this information.

The method for providing proof of a transaction via a network 2 will now be described with reference to a preferred embodiment of the method according to the present invention.

The client who wishes to process an order contacts the web server 15 on the server computer 6 with an Internet browser 10 and establishes a secure connection, e.g. a Secure Socket Layer (SSL) connection. The secure connection may be established using mutual authentication, e.g. using RSA key-pairs and X.509 certificates. After the server computer 6 has been authenticated as a part of the SSL protocol, the client computer 4 needs to identify itself. The RSA key pair of the client resides on the smart card 14 owned by the user. The cryptographic functions needed to establish the connection are preferably executed inside the smart card 14. It is impossible (and not necessary) that the private key leaves the smart card 14. When the client sends his PIN to the smart card 14, this smart card 14 will be enabled to respond to the server challenge to authenticate itself, via the card reader 13, web browser 10 and client network interface 11. A serial number of the smart card 14 will also be part of the client certificate. It is the responsibility of the card-owner to keep the PIN secret.

When a secure connection is established between the client computer 4 and the server computer 6, the client can compose its order. This will be an interactive process between the client and the server. At the end of this process, there will be a concept of the order, ready for the client to accept.

When the concept order is composed, the server computer 6 sends a page to the client computer 4, the page comprising the order content and a confirmation request.

Now, the client has the choice between accepting the order or revoking it. When the order is accepted, the client needs to provide proof of his acceptance and acknowledge the server. This proof is given by processing a digital signature over the page with the order content. The digital signature is computed using the RSA encryption algorithm with the private (secret) key on the smart card 14.

In graphical user-interfaces, which are commonly used in Internet browsers, it is rather easy to tamper with the content of a message, such as the page with the order content. In order to prevent this, the order content is displayed in a tamperproof 'flat text' display. Thus, the client is assured of the correct content of the concept order and no signature will be given for erroneous or fraudulent order content. The text may be made visible to the first client by a text-only dialog on the first client computer, but alternatively, the smart card reader 13 may be provided with an integrated text-only display.

In a next step the client also needs to prepare the smart card 14 to accept a receipt, or Q™ record. For this purpose, the smart card 14 will process a random challenge number that will be used later for the secure write operation to that smart card 14.

The confirmation, the digital signature and the random number will be sent to the server computer 6. An address of the first free position in a Q™ file on the smart card 14 will be sent as well. If there is no free position available, an existing Q™ record needs to be replaced. This can be the oldest Q™ or the Q™ with the lowest priority. The issue date, the expiration date or the priority can be part of the record and will determine which record will be replaced first. The record that is to be replaced can be stored on e.g. a hard disk (not shown) on the client computer 4.

The server computer 6 receives the information as described above. This information, along with the client certificate and the order content, is made available to the Q™ service 18 that is running at the server computer 6.

First, the Q™ service 18 checks the digital signature by means of the client authentication received earlier, a.o. comprising the client public key, and the order content already available in the server computer 6. If the digital signature is right, the service notifies the order application 17 of the fact that the order is confirmed.

The order content, the client information (including public key) and the digital signature will be stored in the Q™ service database 19. This Q™ service database 19 will act as an order administration for the server that stores copies of each order. Each modification of information in this database, be it either to the order content or the digital signature, will be noticed when the digital signatures are checked by means of the public key of the client information.

The smart card 14 is equipped with a Q™ file that is write-protected to prevent unauthorised server computers 6 from writing to the smart card 14. Every smart card 14 has a unique application key that can be computed using a master key in combination with the serial number of the card. The owner of the master key will be able to generate a secure message that can be stored in the Q™ file on the smart card 14.

These messages need to be modified with a random number generated by the smart card 14, as described above. This guarantees that a message can only be written once. If a server computer 6 has the master key, the serial number of the smart card 14

and the random number previously generated by the smart card, this server computer 6 can generate a message that may be saved to the Q™ file on the smart card 14.

The Q™ record that will be stored in the Q™ file on the smart card 14 client may e.g. contain 40 bytes:

-5 bytes unique Q™ number

-16 bytes signed hash of the order content

-19 bytes identification of the server

A special signature key of the server computer 6 signs the 16 bytes hash of the order content. This key is the secret part of a RSA key pair. This key pair can be the same as the authorisation key pair from the server, but it is also possible to use an extra key pair and certificate for this purpose. Of course, the public part of the RSA key pair should be made available to the client computer 4, when different from the public authorisation key. The message will be encrypted using the diversified application key, calculated from the master key and serial number of the smart card 14, and XORed with the random number. The master key resides on the tamperproof SAM (Security Access Module) on the server computer 6, which also may be provided as a smart card and a smart card reader.

The server computer 6 sends a response to the client's browser 10 saying that the order is accepted. This response contains the secure, encrypted message. The encrypted message is sent to the smart card 14 where it will be decrypted and stored.

The client can now read this message (there will be no read protection on this file). The client computer 4 checks whether the signature and the hash on the card correspond with the server's public key as known to the client computer 4 and the hash of the processed order. If this is not the case, the Q™ client service 12 will inform the client that something went wrong and the added Q™ will be deleted.

Normally however, this will not happen. In this case the client computer 4 sends the unique 5-byte Q™ number back to the server. This proofs that the Q™ record is saved on the right smart card 14 of the right client. The Q™ service 18 on the server receives this response and stores in the database 19 that the Q™ record is received on the smart card 14. If this Q™ number does not return, the Q™ service 18 will be able to repeat these steps later.

The estimated time for the complete transaction depends mostly of the response time of the network 2 being used. Using state-of-the-art hardware, the operations

performed on the client computer 4 and the server computer 6 together will not take more than one second. In a further embodiment of the present invention, the server 6 is able to sign by proxy for one or both parties involved in the transaction. In Fig. 2, an embodiment is shown of a further network situation. As discussed with reference to Fig. 1, a first client computer 4 is connected to a web server computer 6 via a network 2 (such as the Internet). The web server computer 6 is connected to a Q™ server computer 25 by means of a direct connection. The web server computer 6 may also be connected to a second client computer 26, e.g. operated by a sales manager. Finally, a trusted third party computer 24 may be connected to the network 2.

With the Q™ system, a user, such as a manager responsible for (part of) the sales via the web server 6, can authorise the Q™ service 18 to sign specific transactions on his behalf, for one or more possible clients. The authorisation may be accomplished directly on the web server computer 6, the Q™ server computer 25 or by means of the second client computer 26. For example, sales manager X can authorise the Q™ service 18 to sign purchase orders from customer Y. Manager X does this by signing a certificate that the Q™ service 18 can then use for signing for transactions between the web server 6 and customer Y, via the customer's client computer 4. Manager X effectively becomes a Certifying Authority for the Q™ service 18. This makes Manager X responsible for what the Q™ service 18 can sign, not the company IT department. The Q™ service 18 simply cannot sign a transaction with customer Z, if there is no certificate from manager X allowing transactions with customer Z.

Of course, it would also be possible that the customer Y sends a certificate to the Q™ service 18, allowing the Q™ service 18 to sign by proxy on behalf of the customer Y, via the client computer 4. This may e.g. be utilised in reverse auction transaction schemes.

The Q™ server 25 can have a certificate that warrants it to sign content for either or both parties. If the Q™ server 25 has to sign for both parties, the originator of the transaction must be verified. The Q™ server 25 must have a warrant that states that the originator (e.g., X or Y) may instruct the server 25 to sign. A policy should be that the originator authorises by way of a Q™ validated order to the server 25.

The proxy certificates the Q™ server 25 has in store must be added by authorised managers, using standard public-key-infrastructure systems, such as X.509 certificates

to validate the both identity of the manager and the correctness of the generated warrant.

The actual proxy signatures that the Q™ server 25 generates must be traceable back to the party that it is signing for. This can either be a simple warrant or a signing-key that is mathematically derived from the original.

A further embodiment of the present invention uses a fair exchange of signatures. This way both participants can be sure that they will receive the signature of the other party, once they commit their signature. This is accomplished by way of a trusted third party on a trusted third party computer 24. This arbitrator *only* needs to be consulted when the exchange fails. Resolving the transaction means that the arbitrator returns a valid Q™ to the Q™ server 25 and/or first client computer 4.

This further method is optional, as not all transaction types require it. Furthermore it requires additional communication and computations that may take too much time or are not possible at all with simple token device.

The transaction process (or Quitum process) will now be explained in more detail with reference to an exemplary embodiment according to the flow diagram shown in Fig. 3.

At the beginning of the Q™ process, in block 301, it is assumed that there is content that needs to be signed. This can be something like a simple order confirmation, a digitally recorded conversation or any other contract.

A customer logs onto a supplier web server 6, and builds an order and requests the order to be shipped. The web server 6 now has a content that needs to be signed by two parties and knows the identity of the customer (Y) and the sales person (X) for the company. The web server 6 now contacts the local Q™ server 25 to start the Q™ process.

There is also a trusted arbitrator for fair signature exchange located in the trusted third party computer 24, which can be contacted by both the first client computer 4 and by the Q™ server 25 (via the web server 6).

In block 302, the Q™ server 25 receives a content to be signed, the identity of the third party Y and the identity of the local signing party X.

In block 303, the Q™ server 25 verifies the identities of X and Y. If either identity cannot be traced back to a trusted Certificate Authority, the Q™ server 25 returns an error.

In its local database, the Q™ server 25 locates the signing key (S) that X has authorised for signing with Y, this way the Q™ server will be signing by proxy for X. S may be a key that has a warrant for only a X(Y) transaction, but the warrant may also state that S may be used for signing with other parties X(Y, Y1, Y2, ... Yn).

The Q™ server now 25 adds to the information a Q™ version number, a timestamp, a transaction ID, and the Transaction Server ID.

Furthermore, the Q™ server 25 now generates or requests a signature for X and creates an ordinary Escrow (non-verifiable) of that signature and attaches to this escrow a description of the signature that Y is supposed to create. A pre-Q™ package is then created from all information except the actual signature for X. This pre-Q™ package is then sent to Y.

In block 304, the first client computer 4 receives the pre-Q™ package and verifies the content. If the customer Y accepts the pre-Q™ package, the first client computer 4 generates its signature for Y and creates a verifiable escrow of this signature in block 305, that is then sent to the Q™ server 25.

The Q™ server 25 verifies the escrow that Y sent in decision block 306 and if it is correct, the Q™ server 25 sends its signature (i.e. the signature of X) to the first client computer 4 in block 308. Otherwise it sends a request to the trusted third party computer 24 to abort the transaction in block 307.

The first client computer 4 receives the signature of X and verifies it in decision block 309. If it is correct, the first client computer 4 in block 311 prepares the secure write to it's token (smart card), as described above with reference to the first embodiment. The first client computer 4 then sends its signature Y and the secure write information to the Q™ server 25.

If the signature of X was incorrect, the first client computer 4 contacts the trusted third party computer 24 and asks it to resolve the transaction in block 310.

At this point, both the first client computer 4 and the Q™ server 25 have essentially signed the content. Only the Q™ server 25 needs to sign one last thing, the receipt it self.

In block 312, the Q™ server 25 can now verify all signatures and, if it is satisfied, sign the receipt itself with its own private key. It will then prepare the secure write to the smart card 14 on the first client computer 4 with its own application key, and send

the result to the first client computer 4. (Block 315, see also the first embodiment described above).

If the signature from the customer Y does not verify, the Q™ server 25 asks the trusted third party computer 24 to resolve the transaction in block 314.

In block 316, the first client computer 4 confirms the secure write by returning the received signature of the Q™ server 25.

At this point the Q™ transaction is complete (block 317).

The customer Y, by means of first client computer 4, can abort the use of the further method using escrows of the signatures by simply sending its signature Y upon acceptance of pre-Q™ package, and not the escrow. The Q™ server 25 cancels the procedure by either immediately sending its signature to the first client computer 4 or by not sending the escrow.

The method using fair exchange makes use of the following additional protocols:

Abort (by originator): The Q™ server 25, which first created a non-verifiable escrow, instructs the trusted third party computer 24 to not resolve this transaction in the future. Neither the customer Y nor the Q™ server 25 can obtain signatures. Only the Q™ server 25 can initiate this request. If the first client computer 4 has already requested a resolve, the Q™ server gets the signature from the customer Y. Only the originator (in the above described example, the Q™ server 25, see block 307) can initiate this sub-protocol.

Resolve (by first client computer 4, see block 310): This resolves the Q™ transaction when the first client computer 4 has sent the verifiable escrow to the Q™ server 25, but has not received a good signature of X back. In exchange for this, the first client computer 4 must deposit the customer's signature Y at the trusted third party computer 24. Because the escrow of signature X is non-verifiable, there is no guarantee that the content of the escrow is correct. If it is not, the resolve will simply fail, any future resolve by the Q™ server 25 will also fail.

Resolve (by Q™ server, see block 314): This resolves the Q™ transaction when the first client computer 4 has not sent a good signature of customer Y to the Q™ server 25 after the Q™ server 25 has sent the signature of manager X. Before doing this, the trusted third party computer 24 verifies that the escrow the first client computer 4 sent to the Q™ server 25 was correct; it does this by checking the condition attached to the verifiable escrow. Only the Q™ server 25 is able to initiate this resolve.

In all the above described embodiments, two main options exist as to what parts of the Q™ information need to be signed by whom. First, all parties sign the Q™ package. In this situation, all parties agree on the content, date/time and identities. Signing of the (order/contract) content is indirect. The other possibility is that the two contract signers only sign the contract, and the Q™ server 25 signs the identities, signatures, content hash and timestamp.

In a first example, the following information is signed in the Q™ process (sample 1) by the customer Y by means of the first client computer 4 and the manager X by means of the Q™ server 25 (sign by proxy):

1. A Q™ version number.
2. A transaction server ID.
3. A transaction ID linked to the server ID from 2.
4. A time stamp.
5. A cryptographically secure hash of the content that has to be signed.
6. The identity of party A.
7. The identity of party B.

The Q™ server 25 signs item 1 through 5. It does not need to sign the identities of either party, as the receipt only has to state that it administered a transaction over the content. It does not have to vouch for the identities of the signers. An added benefit is that the receipt remains relatively small.

As an example, the full Q™ package structure contains the actual Q™ package content with all information:

1. A single byte indicating Q™ version number (currently 1).
2. A transaction (Q™) server ID (SHA fingerprint of the Q™ server RSA1024 bit public key)
3. (4 byte) Transaction ID (server related).
4. The transaction date/time: a Unix 4 byte time structure (seconds since 00:00:00 1970 UTC)
5. A SHA hash of the transaction content (content ID).
6. The SHA fingerprint of the public key of signing party A (as in item 2)
7. The SHA fingerprint of the public key of signing party B (as in item 2)
8. Signature of A over 1 through 7
9. Signature of B over 1 through 7

10. Signature of Q™ server over 1 through 5

A version 1 full Q™ package with 1024 bits signing keys thus amounts to 1+20+4+4+20+20+20+3*(1024/8)= 473 bytes.

The Q™ secure write file is preferably just be a statement from the Q™ server 25. A signature is optional, as the Q™ server is the only one that can write to the file (which implies a signature). The secure write file comprises:

1. Version number
2. Transaction ID
3. Date/time
4. HASH of order content

The result would then be, 1+4+4+20=29 bytes. Note that this file is only valid if it remains on the card.

It will be understood that the above description relates to embodiments and examples related to the present invention. These examples are however not intended to limit the scope of the present invention, which is defined by the claims as attached.

CLAIMS

1. Method for performing a transaction between a first client (4) and a server (6) over a network (2), including the steps of:
providing a server authentication to the first client (4);
providing a first client authentication to the server (6);
sending a record from the server (6) to the first client (4) via the network (2), the record comprising information to be accepted by the first client (4);
providing a first signed record by digitally signing the record by the first client (4);
sending a first message from the first client (4) to the server (6) via the network (2), the first message at least comprising the first signed record; and
storing the record, first message and the first client authentication by the server (6) upon positive verification of the first signed record with the first client authentication, **characterised in that** the method further comprises the steps of
providing a second signed record by digitally signing the record by the server (6);
sending a second message to the first client (4), the second message at least comprising the second signed record; and
storing the record, the second message and the server authentication by the first client (4) upon verification of the second signed record with the server authentication.

2. Method according to claim 1, **characterised in that**
the first client authentication further comprises a smart card serial number,
and in that the method further comprises the steps of
computing a unique application key from the smart card serial number and a master key by the server (6);
using the unique application key to encrypt the second message by the server (6) before sending the second message to the first client (4);
decrypting the second message by the first client (4).

3. Method according to claim 1 or 2, **characterised in that** the first message further comprises a random number,
and in that the method further comprises the step of

performing an exclusive-or-operation by the server (6) on the second message using the random number before sending the second message to the first client (4).

4. Method according to one of the claims 1 through 3, **characterised in that** the step of providing a second signed record comprises the steps of computing a hash of the record and digitally signing the hash of the record by the server (6).

5. Method according to one of the claims 1 through 4, **characterised in that** the step of digitally signing the record by the server (6) is performed using an additional secret server key, the server authentication further comprising an additional public server key.

6. Method according to one of the claims 1 through 5, **characterised in that** the first and second message further comprise a file address for storing the second message by the first client (4).

7. Method according to one of the claims 1 through 6, **characterised in that** the second message further comprises a unique record number,
and in that the method further comprises the step of sending a third message to the server by the first client (4) upon positively verification of the content of the second message, the third message comprising the unique record number.

8. Method according to one of the claims 1 through 7, **characterised in that** the record comprises unformatted text.

9. Method according to one of the claims 1 through 8, **characterised in that** the steps of providing server authentication and first client authentication is obtained by means of RSA-keys and X.509 certificates.

10. Method according to one of the claims 1 through 9, **characterised in that** the second signed record is provided by the server (6) by digitally signing the record using a signing key from a second client (26), the signing key allowing transactions between the second client (26) and at least the first client (4).

11. Method according to claim 10, **characterised in that** the signing key is formed by a certificate signed by the second client (26), the certificate being received from the second client (26) and stored by the server (6).

12 Method according to one of the claims 1 through 11, **characterised in that** the method comprises the further steps of:

sending a non-verifiable escrow of the second signature by the server (6) to the first client (4) before sending the second message; and

sending a verifiable escrow of the first signature by the first client (4) to the server (6) before sending the first message.

13. Method according to claim 12, **characterised in that** the transaction is aborted by the server (6) when the verifiable escrow of the first signature is incorrect.

14. Method according to claim 11 or 12, **characterised in that** the transaction is resolved by a trusted third party (24) when either the second signature or first signature is not correct, by providing the second signature associated with the non-verifiable escrow to the first client (4) or the first signature associated with the verifiable escrow to the server (6).

15. System for performing a secure transaction over a network (2), comprising at least one server computer (6), at least one client computer (4) and a network (2), the network providing a connection between the at least one server computer (6) and the at least one client computer (4), the server computer and client computer being arranged to execute the respective server and first client steps of the method according to one of the claims 1 through 14.

16. System according to claim 15, in which the at least one client computer (4) further comprises memory interface means (13) for communicating with smart memory means (14), the smart memory means (14) being arranged to execute at least one of the first client steps of the method according to one of the claims 1 through 14.

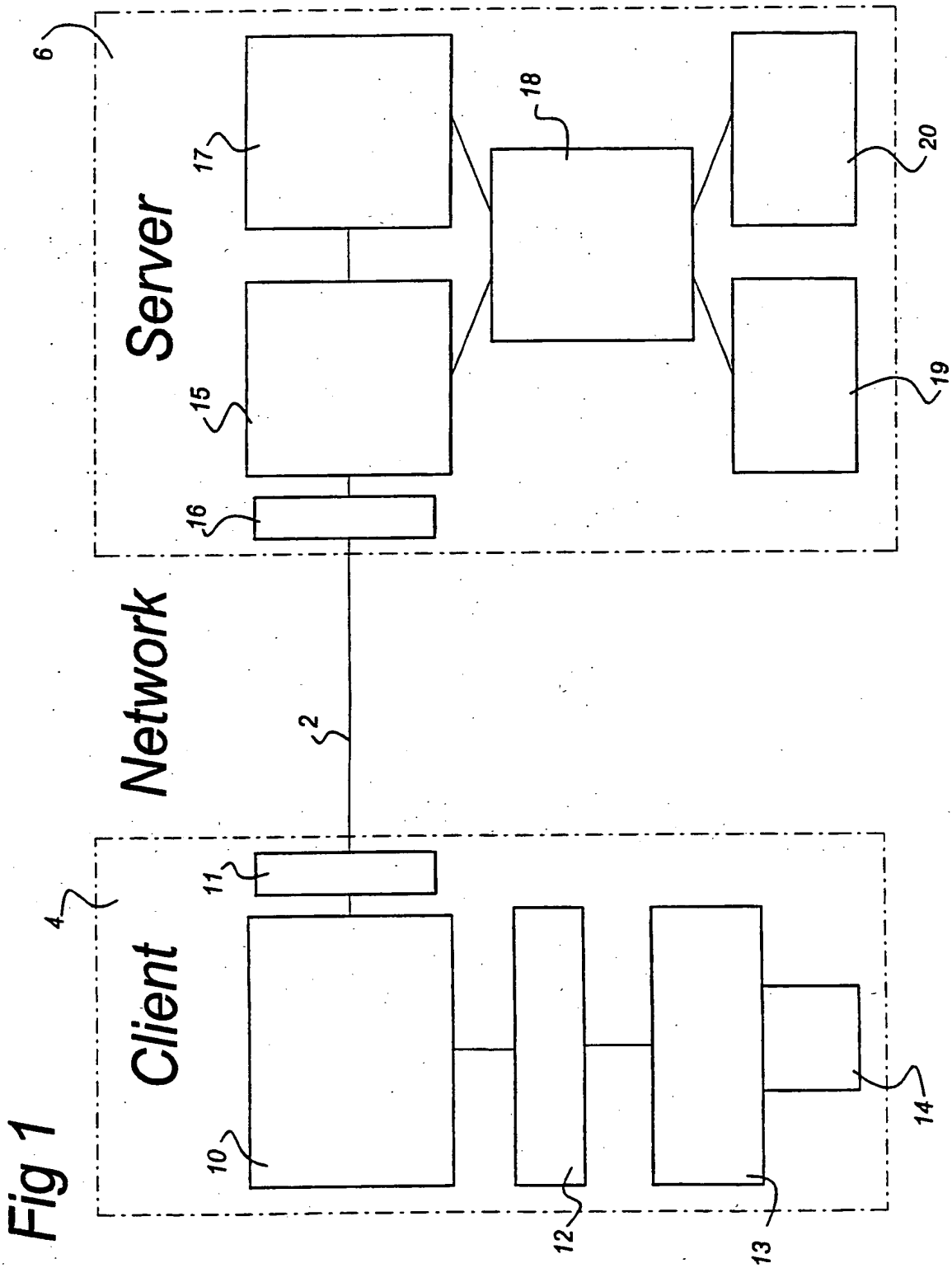


Fig 1

17. System according to claim 15 or 16, in which the system further comprises a trusted third party computer (24) able to communicate with the server computer (6) and the at least one client computer (4).

18. Computer program comprising computer program code, which in operation provides a computer system with the functionality of the method according to one of the claims 1 through 14.

19. Computer program product comprising the computer program of claim 18.

****

Fig 2

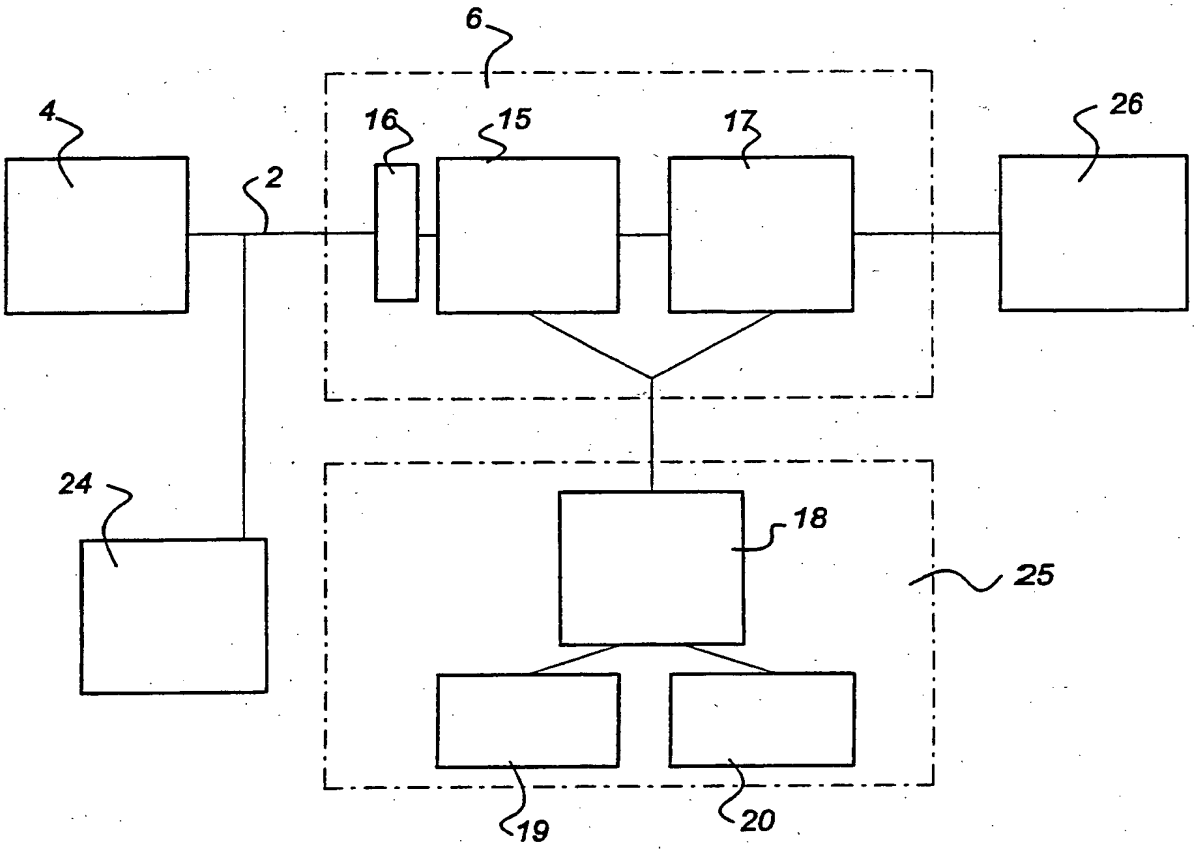
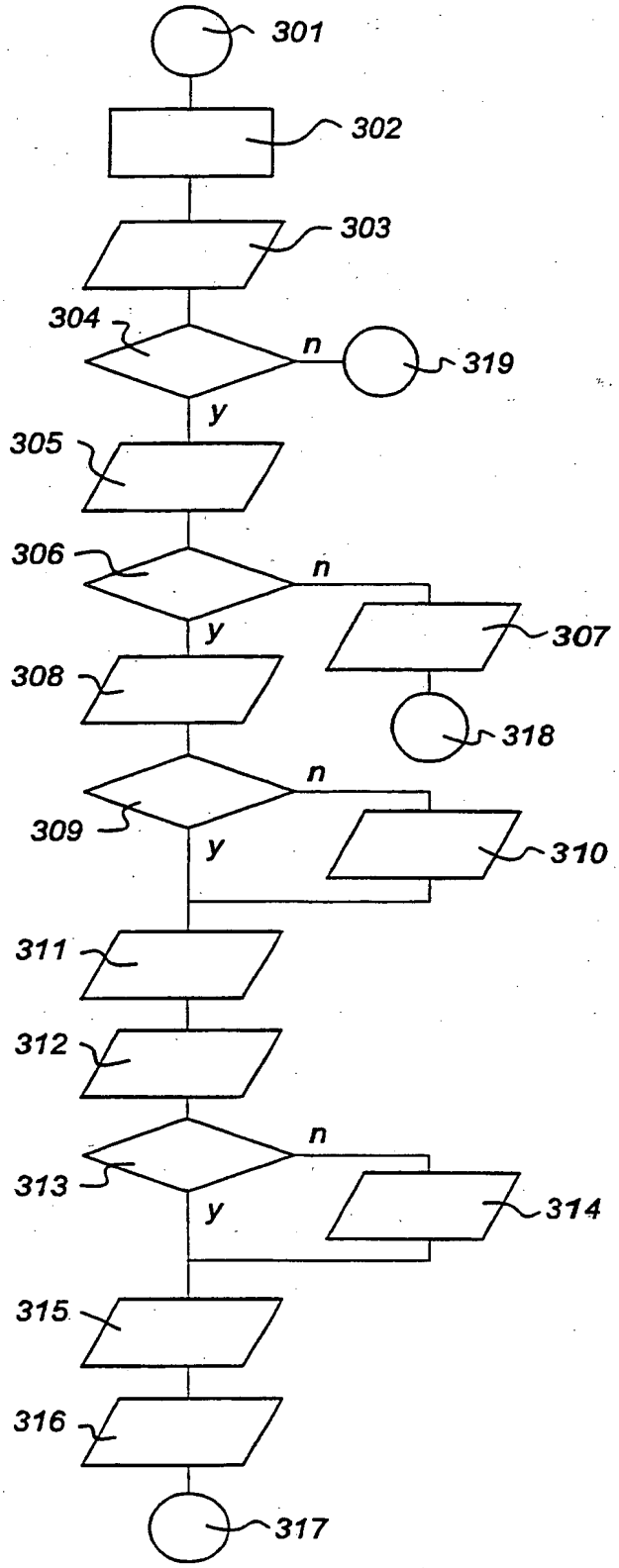


Fig 3



INTERNATIONAL SEARCH REPORT

Inter. nal Application No
PCT/NL 00/00217

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G07F7/10 G07F19/00 H04L9/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G07F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 018 196 A (K. TAKARAGI) 21 May 1991 (1991-05-21)	1,4,5,9
A	abstract; figures 1,2 column 4, line 14 -column 5, line 14 column 7, line 8 -column 8, line 68	2,7, 12-19
A	WO 96 31965 A (FINANCIAL SERVICES TECHNOLOGY CONSORTIUM) 10 October 1996 (1996-10-10) abstract; claims; figure 3 page 14, line 17 -page 18, line 18	1,5,9, 15-19
A	WO 93 08545 A (JONHIG) 29 April 1993 (1993-04-29) abstract; claims; figures page 11, line 8 -page 13, line 13	1-5,7,9, 15,16,18

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 June 2000	29/06/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	David, J

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter nal Application No
PCT/NL 00/00217

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 172 670 A (TECHNION RESEARCH & DEVELOPMENT FOUNDATION) 26 February 1986 (1986-02-26) abstract; claims; figures column 6, line 49 - line 56 column 9, line 5 - line 32	1,5,9, 15,16,18
A	DE 44 27 039 A (GIESECKE & DEVRIENT) 1 February 1996 (1996-02-01)	
A	US 5 577 121 A (T.L. DAVIS) 19 November 1996 (1996-11-19)	
A	US 5 809 144 A (M.A. SIRBU) 15 September 1998 (1998-09-15)	

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

..formation on patent family members

Interi nal Application No
PCT/NL 00/00217

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5018196 A	21-05-1991	JP 2112794 C	21-11-1996
		JP 8027812 B	21-03-1996
		JP 62254543 A	06-11-1987
		JP 62056043 A	11-03-1987
		JP 2170184 A	29-06-1990
		US 4885777 A	05-12-1989
		DE 3687934 A	15-04-1993
		DE 3687934 T	17-06-1993
		EP 0214609 A	18-03-1987
WO 9631965 A	10-10-1996	US 5677955 A	14-10-1997
		BR 9608448 A	07-12-1999
		CA 2217593 A	10-10-1996
		EP 0819345 A	21-01-1998
		JP 11503541 T	26-03-1999
WO 9308545 A	29-04-1993	AT 145744 T	15-12-1996
		AU 663739 B	19-10-1995
		AU 2888692 A	21-05-1993
		BR 9205416 A	17-05-1994
		CA 2098481 A	17-04-1993
		DE 69215501 D	09-01-1997
		DE 69215501 T	27-03-1997
		DK 567610 T	17-02-1997
		EP 0567610 A	03-11-1993
		ES 2096772 T	16-03-1997
		GR 3022528 T	31-05-1997
		HK 1001573 A	26-06-1998
		JP 2853331 B	03-02-1999
		JP 6503913 T	28-04-1994
		KR 161670 B	20-03-1999
		MD 1402 F	31-01-2000
		NO 303893 B	14-09-1998
		PL 299825 A	18-04-1994
		US 5440634 A	08-08-1995
EP 0172670 A	26-02-1986	JP 61094177 A	13-05-1986
DE 4427039 A	01-02-1996	EP 0696021 A	07-02-1996
US 5577121 A	19-11-1996	US 5892211 A	06-04-1999
US 5809144 A	15-09-1998	NONE	